Assignment 1 – PROLOG part

Q3. Unification:

1. Error: functor Food(Y,soup) cannot start with an upper case letter.
2. Yes: Bread = soup.
3. Yes: For any case where Bread and Soup holds the same value.
4. No because in LHS, X should be salad, but on RHS, X should be milk.
5. No because the arity on LHS is not the same as the arity on RHS
6. Yes: X = healthyFood(bread) and Y = drink(milk)
8. Error: The list on the RHS has a misuse of “I” operator
9. Can be unified: X = l and Z =b
10. Yes: A = french(jean) and B = scottish(joe)
11. Yes: Y = drink(water) and X = healthyFood(bread)
12. Yes: H = a and T = [b, c]
13. No because the two lists are not of the same size.
14. Yes: Only if healthyFood(egg) unifies with healthyFood(bread). Then, Y=egg and Z = milk
15. Yes: X= jack, Y = cook(egg,oil), Time = Evening)
16. Yes: X = s(g) and Y = t(k)
17. Yes: Only if f(x,17,M) = f(x,x,M) = f(17,17,M), then Z = C, D = 17 or D = x, C = L\*y, E = 17
18. No: b is not a list, so it can’t be unified with [H|T]

Q4.

1. Ground
2. The engine will try to match the query one clause at a time starting from the top and going to the bottom until it is able to unify with the rule field(X,Y).
3. Prolog will instantiate X to heat\_transfer and Y to engineering.
4. Resolution will apply the substitution of variable in the rule, therefore creating a new rule: field(heat\_transfer,engineering):-course(heat\_transfer, Z), field(Z,engineering).
5. The two new goals will be processed and the engine will try to unify them for a certain value of Z. In this case, Prolog will find that the two goals are validated with Z = mechanical.
6. The engine will therefore return **true**.
7. Non-Ground
8. Ground
9. Non-Ground
10. Ground
11. Ground
12. Non-Ground
13. Non-Ground
14. Non-Ground
15. Non-Ground
16. Non-Ground
17. Non-Ground
18. Non-Ground
19. Non-Ground